An augmented approach for Stokes equations with discontinuous viscosity and singular forces

Zhilin Li[*]   Kuzufumi Ito[†]   Ming-Chih Lai[‡]

Abstract

For Stokes equations with discontinuous viscosity across an arbitrary interface or/and singular forces along the interface, the pressure is known to be discontinuous and the velocity is known to be non-smooth. It has been shown that these discontinuities are coupled together which makes it difficult to obtain accurate numerical solutions. In this paper, a second order accurate numerical method that decouples the jump conditions of the fluid variables through two augmented variables has been developed. The GMRES iterative method is used to solve the Schur complement system for the augmented variables which are only defined on the interface. The augmented approach also rescales the Stokes equations in such a way that fast Poisson solvers can be used in each iteration. Numerical examples against exact solutions show that the new method has average second order accuracy in the infinity norm, and the number of GMRES iterations is independent of mesh sizes. An example of a moving interface problem is also presented.

Keywords: Incompressible Stokes flow, discontinuous viscosity, coupled jump conditions, interface, discontinuous and non-smooth solution, immersed interface method, fast Poisson solver

AMS Classification; 65M06, 65M12, 76T05

1   Introduction

The incompressible Stokes or Navier-Stokes equations with discontinuous viscosity and singular forces arise from many important applications in fluid and biofluid mechanics. One of particular example is Peskin's immersed boundary (IB) model to simulate the blood flow in a

---

[*]Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA. E-mail: zhilin@math.ncsu.edu

[†]Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA. E-mail: kito@math.ncsu.edu

[‡]Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan. E-mail: mclai@math.nctu.edu.tw

1. REPORT DATE **2004**	2. REPORT TYPE	3. DATES COVERED **00-00-2004 to 00-00-2004**
4. TITLE AND SUBTITLE **An augmented approach for Stokes equations with discontinuous viscosity and singular forces**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **North Carolina State University,Center for Research in Scientific Computation,Raleigh,NC,27695-8205**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT **see report**		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **23**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

human's heart [19, 20, 23]. The idea of IB formulation is to treat the complicated immersed boundary (such as a heart valve) as a force generator in the fluid domain, or mathematically, a Dirac delta function force distribution along the immersed boundary. While Peskin's IB model has been intensively studied in the literature, almost none has studied the case where the viscosity is discontinuous. In this paper, we consider the following two-dimensional stationary incompressible Stokes equations

$$\nabla p = \nabla \cdot \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) + \mathbf{g} + \int_\Gamma \mathbf{f}(s) \, \delta_2(\mathbf{x} - \mathbf{X}(s)) ds, \quad \mathbf{x} \in \Omega, \qquad (1.1)$$

$$\nabla \cdot \mathbf{u} = \mathbf{0}, \quad \mathbf{x} \in \Omega, \qquad (1.2)$$

where $\mathbf{u} = (u, v)$ is the velocity vector, p is the pressure, Γ is an arbitrary interface parameterized by the arc-length s, $\mathbf{f}$ is the density of the force strength along Γ[1], μ is the viscosity, which we assume to be a piecewise constant

$$\mu = \begin{cases} \mu^+, & \text{if } \mathbf{x} \in \Omega^+, \\ \mu^-, & \text{if } \mathbf{x} \in \Omega^-, \end{cases} \qquad (1.3)$$

with a finite jump across Γ, $\mathbf{g}(\mathbf{x})$ is a bounded forcing term, *e.g.*, the gravitational force, which can be discontinuous across Γ as well, and Ω is a bounded domain which we assume to be a rectangle, see Fig. 1 (a) for an illustration of the problem. We assume periodic boundary conditions for the pressure and the velocity in this paper. The system (1.1)-(1.3) is the <u>model equations</u> that we are going to discuss in this paper. The existence of the solution to the system (1.1)-(1.2) can be found in [22].

There are at least two difficulties in solving (1.1)-(1.2) numerically. The first one is to deal with the singular force term. A simple way is to use Peskin's discrete delta function approach to distribute the singular force to nearby grid points. Such a discretization is typically first order accurate and will smooth out the solution. The second difficulty is how to deal with the discontinuous viscosity. A simple smoothing method may have large errors, see [15] for a one dimensional example there.

In the case of the continuous viscosity, various methods have been developed in the literature. We refer the readers to [3, 4, 12, 18, 23] for various methods and the references therein. The difficulty with discontinuous viscosity is that the jump conditions for the pressure and velocity are coupled together, see (2.4)-(2.7), which makes it difficult to discretize the system accurately.

In this paper, we develop a new second-order sharp interface method which uses the exact jump conditions for two-phase incompressible Stokes flow with discontinuous viscosity. We believe that our method is the first sharp interface method with second-order accuracy. The

---

[1]The singular source term $\int_\Gamma \mathbf{f}(s) \, \delta_2(\mathbf{x} - \mathbf{X}(s)) ds$ can also be written as $((\mathbf{f} \cdot \mathbf{n})\mathbf{n} + (\mathbf{f} \cdot \boldsymbol{\tau})\boldsymbol{\tau}) \, \delta(\Gamma)$, or in the form of $((\mathbf{f} \cdot \mathbf{n})\mathbf{n} + (\mathbf{f} \cdot \boldsymbol{\tau})\boldsymbol{\tau}) \, |\nabla\varphi| \, \delta(\varphi)$, where φ is a level set function whose zero level set represents the interface Γ.
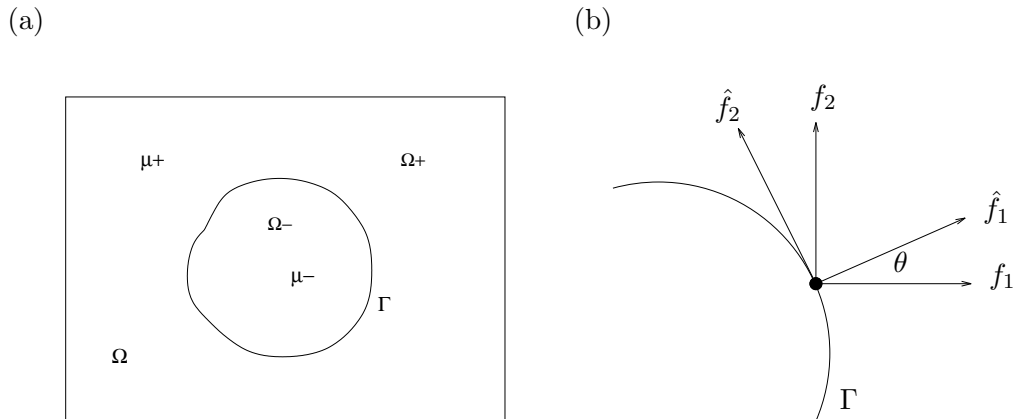
(a)   (b)

Figure 1: (a): A diagram for the incompressible Stokes equations defined on a domain Ω with an interface Γ across which the viscosity μ is discontinuous. (b): Force density decomposition in which f_1 and f_2 are the force density in the x- and y- directions, while $\hat{f}_1$ and $\hat{f}_2$ are the force density in the normal and tangential directions.

idea is to introduce two augmented variables that are defined only along the interface so that the jump conditions can be decoupled and the immersed interface method [11, 13] can be applied. The GMRES iterative method is applied to the Schur complement system for the discrete augmented variables. Furthermore, our approach rescales the original problem and enables us to use fast Poisson solvers in the iterative process. Each GMRES iteration requires to solve the rescaled Stokes equations, which can be done by calling a fast Poisson solver three times, and an interpolation scheme to evaluate the residual of the Schur complement.

We should mention other related work in this area. One is the fast Stokes solver on irregular domains based on an integral equation approach [1, 5] which may be applied to the two-phase Stokes flow discussed here. There are also some work on the Navier-Stokes equations with discontinuous viscosity, for example, the first order accurate ghost fluid method [9]. The system of Navier-Stokes equations is a time dependent problem in which we can use a time marching method. The Stokes flow discussed here, on the other hand, is an elliptic system in which we have to solve the entire system simultaneously.

The paper is organized as follows. In the next section, we give some discussions on the jump conditions of the Stokes equations involving an interface. We will see how the jump conditions for the pressure and velocity are coupled together, and explain the idea how to decouple the jump conditions by introducing two augmented variables and two augmented equations. In Sec. 3, we present the new algorithm in detail. We explain how the GMRES iterative method can be applied to the discrete augmented variables without explicitly form the coefficient matrix. In Sec. 4, we present some numerical experiments against the exact solutions to check the performance of new method. An example of moving interface is also presented there.

2 Jump conditions for incompressible two-phase Stokes equations with discontinuous viscosity and singular forces

Referring to Fig 1 (a) and the equations (1.1)-(1.3), we assume the interface Γ is a smooth curve. At a point (X, Y) on the interface, we use the notations $\mathbf{n}$ and $\boldsymbol{\tau}$ to represent the unit normal and tangential directions respectively. We use $\hat{f}_1(s) = \mathbf{f}(s) \cdot \mathbf{n}$ and $\hat{f}_2(s) = \mathbf{f}(s) \cdot \boldsymbol{\tau}$ to represent the force density in the normal and tangential directions. If Γ is smooth, then in a neighborhood of Γ, the distance function $d(\mathbf{x}, \Gamma)$ is also a smooth function. The normal and tangential directions of Γ can be extended to the neighborhood, for example $\mathbf{n} = \nabla d(\mathbf{x}, \Gamma)/|\nabla d(\mathbf{x}, \Gamma)|$. Therefore the quantities such as $\mathbf{u} \cdot \mathbf{n}$, $\frac{\partial \mathbf{u}}{\partial n}$, $\frac{\partial \mathbf{u}}{\partial \tau}$, etc., are well defined in the neighborhood of Γ.

In [8], the interface conditions for two-phase Stokes equations with discontinuous viscosity and a Dirac delta function singularity are derived which is summarized in the following theorem:

Theorem 1 *Assume $\Gamma(s) \in C^2$, $\hat{f}_1(s) \in C^1$, and $\hat{f}_2(s) \in C^1$. Let p and $\mathbf{u} = (u, v)$ be the solution to the Stokes equations (1.1)-(1.2). We have the following jump conditions across the interface Γ.*

$$[p] = 2 \left[\mu \frac{\partial \mathbf{u}}{\partial n} \cdot \mathbf{n} \right] + \hat{f}_1, \tag{2.4}$$

$$\left[\frac{\partial p}{\partial n} \right] = [\mathbf{g} \cdot \mathbf{n}] + \frac{\partial}{\partial \tau} \hat{f}_2 + 2 \left[\mu \frac{\partial^2}{\partial \tau^2} (\mathbf{u} \cdot \mathbf{n}) \right], \tag{2.5}$$

$$\left[\mu \frac{\partial \mathbf{u}}{\partial n} \cdot \boldsymbol{\tau} \right] + \left[\mu \frac{\partial \mathbf{u}}{\partial \tau} \cdot \mathbf{n} \right] + \hat{f}_2 = 0, \tag{2.6}$$

$$[\mu \nabla \cdot \mathbf{u}] = 0. \tag{2.7}$$

where the jump $[\,\cdot\,]$ of a quantity, for example, $[p]$ at a point $\mathbf{X}$ is defined as

$$[p] \Big|_{\mathbf{X} \in \Gamma} \overset{\text{def}}{=} \lim_{\mathbf{x} \to \mathbf{X}, \mathbf{x} \in \Omega^+} p(\mathbf{x}) - \lim_{\mathbf{x} \to \mathbf{X}, \mathbf{x} \in \Omega^-} p(\mathbf{x}). \tag{2.8}$$

We will omit the subscript $\mathbf{X} \in \Gamma$ in the rest of the paper for simplicity of the notations. The first jump condition (2.4) is the result of balancing force in the normal direction while the third one (2.6) is the result of balancing force in the tangential direction. The last jump condition (2.7) is obtained from the incompressibility condition. The jump condition for the normal derivative of the pressure (2.5) can be obtained by applying the divergence operator to the momentum equation (1.1), see [8]. It is worthy mentioning that the equation (1.1)-(1.2) can be re-written as the one without the singular force term but accompanied with the above jump conditions.

From the incompressibility condition, one can easily prove that $[\frac{\partial \mathbf{u}}{\partial n} \cdot \mathbf{n}] = 0$. If μ is continuous, then $[\mu \frac{\partial \mathbf{u}}{\partial n} \cdot \mathbf{n}] = 0$ and $[\mu \frac{\partial^2}{\partial \tau^2}(\mathbf{u} \cdot \mathbf{n})] = 0$, and the jump conditions for the pressure and the velocity are decoupled from (2.4)-(2.7). In this case, we recover the jump conditions derived and used in [12, 10]. The second order accurate immersed interface method has been developed in [12, 13] if the viscosity is continuous. When μ has a finite jump across the interface Γ, to our best knowledge, there is not yet second order accurate sharp interface method in the literature for the Stokes equations. To get a second order accurate algorithm based on the immersed interface method, our strategy is to introduce two intermediate, or augmented variables, along the interface so that the jump conditions can be decoupled. Meanwhile, we also need two augmented equations to complete the system of the governing equations.

There are more than one ways to introduce augmented variables so that the jump conditions can be decoupled. Different augmented variables and equations will lead to different algorithms. Note that for discontinuous viscosity, there are at least two different scales corresponding to the two-phase flow. Furthermore, with constant viscosity, we can use fast Poisson solvers to get the solution of the Stokes flow [12, 13]. Based on these two considerations, we introduce the jumps $[\mu u](s)$ and $[\mu v](s)$ along the interface as two augmented variables. The advantages and details can be seen in the rest of the paper.

Before we proceed, we introduce the local coordinates transform at a point (X, Y) on the interface Γ as,

$$\begin{cases} \xi = (x - X)\cos\theta + (y - Y)\sin\theta, \\ \eta = -(x - X)\sin\theta + (y - Y)\cos\theta, \end{cases} \tag{2.9}$$

where θ is the angle between the x-axis and the normal direction at the point (X, Y), see Fig. 1 (b). Using this local coordinate system, we can rewrite the last two jump conditions (2.6)-(2.7) in terms of the augmented variables $[\mu u](s)$ and $[\mu v](s)$ as follows.

Lemma 1 *Let p, u, and v be the solution to the Stokes equation (1.1)-(1.2). Define*

$$\tilde{u} = \mu u, \qquad \tilde{v} = \mu v, \qquad \tilde{\mathbf{u}} = (\tilde{u}, \tilde{v}). \tag{2.10}$$

Then we have the following jump relations for $\tilde{u}$ and $\tilde{v}$

$$\left[\frac{\partial \tilde{u}}{\partial n}\right] = \left(\hat{f}_2 + \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \mathbf{n}\right]\right)\sin\theta - \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right]\cos\theta, \tag{2.11}$$

$$\left[\frac{\partial \tilde{v}}{\partial n}\right] = -\left(\hat{f}_2 + \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \mathbf{n}\right]\right)\cos\theta - \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right]\sin\theta. \tag{2.12}$$

$$\left[\frac{\partial \tilde{\mathbf{u}}}{\partial n} \cdot \mathbf{n}\right] = -\left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right] \tag{2.13}$$

Proof: Note that $\mathbf{n} = (\cos\theta, \sin\theta)$ and $\boldsymbol{\tau} = (-\sin\theta, \cos\theta)$. Re-write the incompress-

ibility condition $[\mu \nabla \cdot \mathbf{u}] = 0$ in the local coordinates, we have

$$\left[\frac{\partial \tilde{u}}{\partial n}\right] \cos \theta - \left[\frac{\partial \tilde{u}}{\partial \tau}\right] \sin \theta + \left[\frac{\partial \tilde{v}}{\partial n}\right] \sin \theta + \left[\frac{\partial \tilde{v}}{\partial \tau}\right] \cos \theta = 0, \tag{2.14}$$

which is

$$\left[\frac{\partial \tilde{u}}{\partial n}\right] \cos \theta + \left[\frac{\partial \tilde{v}}{\partial n}\right] \sin \theta = - \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right]. \tag{2.15}$$

Re-write the interface relation (2.6) in the local coordinates, we have

$$- \left[\frac{\partial \tilde{u}}{\partial n}\right] \sin \theta + \left[\frac{\partial \tilde{v}}{\partial n}\right] \cos \theta = -\hat{f}_2 - \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \mathbf{n}\right]. \tag{2.16}$$

From the two equalities (2.15) and (2.16) above, we solve for $[\frac{\partial \tilde{u}}{\partial n}]$ and $[\frac{\partial \tilde{v}}{\partial n}]$ to get (2.11) and (2.12). The last equality is verified by substituting $[\frac{\partial \tilde{u}}{\partial n}]$ with (2.11), and $[\frac{\partial \tilde{v}}{\partial n}]$ with (2.12) in the following

$$\left[\frac{\partial \tilde{\mathbf{u}}}{\partial n} \cdot \mathbf{n}\right] = \left[\frac{\partial \tilde{u}}{\partial n}\right] \cos \theta + \left[\frac{\partial \tilde{v}}{\partial n}\right] \sin \theta = - \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right].$$

3 The numerical algorithm

Our numerical method is based on the following theorem.

Theorem 2 *Let p, u, and v be the solution to the Stokes equations (1.1)-(1.2). Let $q_1(s) = [\tilde{u}](s) = [\mu u](s)$, $q_2(s) = [\tilde{v}](s) = [\mu v](s)$, and $\mathbf{q}(s) = (q_1(s), q_2(s))$. Then $\tilde{u}$, $\tilde{v}$, p, $q_1(s)$, $q_2(s)$ are the solution of the following augmented system of partial differential equations:*

$$\begin{cases} \Delta p = \nabla \cdot \mathbf{g}, \\ [p] = \hat{f}_1 - 2\dfrac{\partial \mathbf{q}}{\partial \tau} \cdot \boldsymbol{\tau}, \qquad \left[\dfrac{\partial p}{\partial n}\right] = \dfrac{\partial \hat{f}_2}{\partial \tau} + 2\dfrac{\partial^2 (\mathbf{q} \cdot \mathbf{n})}{\partial \tau^2}, \end{cases} \tag{3.17}$$

$$\begin{cases} \Delta \tilde{u} = p_x - g_1, \\ [\tilde{u}] = q_1, \qquad \left[\dfrac{\partial \tilde{u}}{\partial n}\right] = \left(\hat{f}_2 + \dfrac{\partial \mathbf{q}}{\partial \tau} \cdot \mathbf{n}\right) \sin \theta - \left(\dfrac{\partial \mathbf{q}}{\partial \tau} \cdot \boldsymbol{\tau}\right) \cos \theta, \end{cases} \tag{3.18}$$

$$\begin{cases} \Delta \tilde{v} = p_y - g_2, \\ [\tilde{v}] = q_2, \qquad \left[\dfrac{\partial \tilde{v}}{\partial n}\right] = -\left(\hat{f}_2 + \dfrac{\partial \mathbf{q}}{\partial \tau} \cdot \mathbf{n}\right) \cos \theta - \left(\dfrac{\partial \mathbf{q}}{\partial \tau} \cdot \boldsymbol{\tau}\right) \sin \theta, \end{cases} \tag{3.19}$$

$$\left[\frac{\tilde{u}}{\mu}\right] = 0, \qquad \left[\frac{\tilde{v}}{\mu}\right] = 0. \tag{3.20}$$

The proof of the theorem is straightforward from the Stokes equations (1.1)-(1.2), the jump conditions in Theorem 1 and Lemma 1. The periodic boundary condition is used so that we are not introducing additional boundary condition for the pressure.

The existence and uniqueness of the solution to the system above is the same as the original incompressible Stokes equations (1.1)-(1.2). This is because if (u, v) and p are the solution to the original Stokes equations, then they are also the solution to the system above according to the definition of (u, v), Theorem 1, and Lemma 1. On the other hand, if (u, v) and p are the solution to the system (3.17)-(3.20) above plus the periodic boundary condition, then they satisfy all the equations in (1.1)-(1.2) and the incompressibility condition. So they are also solution to the original problem.

Notice that if we know $\mathbf{q}$, then the jump conditions for the pressure are all known and we can solve the pressure independent of the velocity. After the pressure is solved, we can solve the velocity from (3.18) and (3.19). The three equations with the given jump conditions can be solved using the immersed interface method [11, 12, 13] in which a fast solver is called with modified right hand sides at grid points near or on the interface. This observation is the basis of our new method. We can try to find $\mathbf{q}$ iteratively starting from an initial guess. The compatibility condition for the two augmented variables are the two equations in (3.20) (which means that the velocity is continuous across the interface). It is also important to mention that the incompressibility condition is used to obtain the pressure Poisson equation of (3.17).

Once the augmented variables ($[\mu u]$ and $[\mu v]$) and the augmented equations (the two equations in (3.20)) are chosen, the success of the numerical algorithm depends on how efficiently we can solve the augmented variables. Note that the augmented approaches have been developed for elliptic interface problems with piecewise constant coefficient [6, 14], and the fast algorithms for Poisson and biharmonic equations on irregular domains [2, 7, 17, 16]. However, the augmented approach proposed here is for a more difficult problem and it may be the only to get a second order sharp interface method to solve the Stokes flow with discontinuous viscosity. The key to the success of an augmented approach is the choice of the augmented variable(s) and equation(s) which is a research process and depends on problems.

We assume that the domain Ω is a rectangle: $[a, b] \times [c, d]$. We use a uniform Cartesian grid

$$x_i = a + ih_x, \quad i = 0, 1, \cdots, m, \quad h_x = \frac{b - a}{M},$$

$$y_j = c + jh_y, \quad j = 0, 1, \cdots, n, \quad h_y = \frac{d - c}{N},$$

so that a fast Poisson solver can be used to solve the three equations (3.17)-(3.19).

We first choose a set of control points $\{\mathbf{X}_k\} = \{(X_k, Y_k)\}$, $k = 1, 2, \cdots, N_b$, on the interface[2]. The auxiliary variable $\mathbf{q}(s) = (q_1(s), q_2(s))$ is defined, and the augmented equations (3.20) are discretized, at $\{\mathbf{X}_k\}$. We use upper case letters such as P_{ij}, U_{ij}, V_{ij}, $\mathbf{Q}_k$, for the discrete approximations at grid points and at those control points on the interface, respectively. We use the bold face upper case letters without subscripts to represent the vectors formed by those discrete components.

Given an initial guess of $\mathbf{Q}$ at the control points, we can approximate its first and second order tangential derivatives $\frac{\partial \mathbf{q}}{\partial \tau}$ and $\frac{\partial^2 \mathbf{q}}{\partial \tau^2}$, see [11, 13]. Thus all the jump conditions in the system (3.17)-(3.19) are known and we can solve the system (3.17)-(3.19) sequentially using the immersed interface method [11, 12]. Note that the jump conditions for the pressure and the velocity now are decoupled if we know $\mathbf{Q}$. Since the solution depends on $\mathbf{Q}$, the solution can be written as $\mathbf{P}(\mathbf{Q})$, $\tilde{\mathbf{U}}(\mathbf{Q})$ and $\tilde{\mathbf{V}}(\mathbf{Q})$.

If the computed $\tilde{\mathbf{U}}(\mathbf{Q})$ and $\tilde{\mathbf{V}}(\mathbf{Q})$ satisfy the two equations in (3.20), then $\mathbf{P}(\mathbf{Q})$, $\tilde{\mathbf{U}}(\mathbf{Q})/\mu$ and $\tilde{\mathbf{V}}(\mathbf{Q})/\mu$ are an approximated solution to the original system (1.1)-(1.2). Otherwise, we use a third order accurate linear interpolation scheme to evaluate the residual of the two equations in (3.20) which will be explained in Section 3.2.

The core of our algorithm is to use the GMRES iterative method [21] to solve for the augmented variable $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ which are one dimensional vectors defined along the interface Γ, or more precisely, at the control points $\{\mathbf{X}_k\}$. However, there is no need to find the coefficient matrix explicitly as we will explain below. Each GMRES iteration requires to solve three Poisson equations with given jumps in the solution and its normal derivative, and a least squares interpolation scheme to evaluate the residual. It is worth to point out that the most expensive part of the algorithm, which is to solve the three Poisson equations, can be done by calling a fast Poisson solver three times.

3.1 The discrete system of equations in the matrix vector form

Given a discrete approximation of (q_1, q_2) at $\{\mathbf{X}_k\}$, we can solve the first three equations (3.17)-(3.19) using the immersed interface method [12] to get an approximate solution: the pressure $\mathbf{P}(\mathbf{Q})$, the scaled velocity $\tilde{\mathbf{U}}(\mathbf{Q})$ and $\tilde{\mathbf{V}}(\mathbf{Q})$. Generally the computed velocity $(\tilde{\mathbf{U}}(\mathbf{Q}), \tilde{\mathbf{V}}(\mathbf{Q}))$ do not satisfy the two augmented equations in (3.20), that is, $(\mathbf{U}, \mathbf{V}) = (\tilde{\mathbf{U}}/\mu, \tilde{\mathbf{V}}/\mu)$ may not be continuous across the interface.

Let us put the discrete solution $\{P_{ij}\}$, $\{U_{ij}\}$, and $\{V_{ij}\}$ together as a big vector $\tilde{\mathcal{U}}$ whose dimension is $3MN$. We denote also the vector of the discrete values of (q_1, q_2) at the control points $\{\mathbf{X}_k\}$ by $\mathbf{Q}$ whose dimension is $2N_b$. Then the discrete solution of (3.17)-(3.19) given

---

[2]In a front tracking method, $\{\mathbf{X}_k\}$ is a set of points that represents the interface, see [13] for examples in which a cubic spline is used. In a level set method, the control points can be taken as the orthogonal projections of irregular grid points on the interface, see [6, 14] for example.

Q can be written as

$$A\tilde{\mathcal{U}} + B\mathbf{Q} = \mathbf{F}_1 \tag{3.21}$$

for some vector $\mathbf{F}_1$ and sparse matrices A and B. It requires solving three Poisson equations with different force terms and jump conditions to get $\tilde{\mathcal{U}}$.

Once we know the solution $\tilde{\mathcal{U}}$ given **Q**, we can use $(\tilde{U}, \tilde{V})$ and the jump conditions $[\frac{\partial \tilde{U}}{\partial n}]$ and $[\frac{\partial \tilde{V}}{\partial n}]$ which also depend on **Q**, to get $[\mathbf{U}(\mathbf{Q})] = [\tilde{U}(\mathbf{Q})/\mu]$ and $[\mathbf{V}(\mathbf{Q})] = [\tilde{V}(\mathbf{Q})/\mu]$ at those control points $\{\mathbf{X}_k\}$, $1 \leq k \leq N_b$. If both $\|[\mathbf{U}(\mathbf{Q})]\|$ and $\|[\mathbf{V}(\mathbf{Q})]\|$ are smaller than a given tolerance, then the method has already converged and **Q**, $\tilde{U}/\mu$, $\tilde{V}/\mu$ are the approximate solution. The interpolation scheme to get $[\tilde{U}(\mathbf{Q})/\mu]$ and $[\tilde{V}(\mathbf{Q})/\mu]$, which will be explained in detail in the next sub-section, is *linearly* dependent of $\tilde{\mathcal{U}}$, **Q**. Therefore we can write

$$[\mathbf{U}(\mathbf{Q})]\Big|_{\Gamma} = \left(\left[\tilde{U}/\mu \right], \left[\tilde{U}/\mu \right] \right)^T = S\tilde{\mathcal{U}} + E\mathbf{Q} - \mathbf{F}_2, \tag{3.22}$$

where S and E are two sparse matrices, and $\mathbf{F}_2$ is a vector. The matrices depend on the interpolation scheme and are only used for theoretical purpose here but not actually constructed in our algorithm. We need to choose such a vector **Q** that the continuity condition for the velocity is satisfied along the interface Γ. If we put the two matrix-vector equations (3.21) and (3.22) together we get

$$\begin{bmatrix} A & B \\ S & E \end{bmatrix} \begin{bmatrix} \tilde{\mathcal{U}} \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{bmatrix}. \tag{3.23}$$

Note that **Q** is defined only on a set of points $\{\mathbf{X}_k\}$ on the interface while $\tilde{\mathcal{U}}$ is defined at all grid points. The Schur complement for **Q** is

$$(E - SA^{-1}B)\mathbf{Q} = \mathbf{F}_2 - SA^{-1}\mathbf{F}_1 = \bar{\mathbf{F}}. \tag{3.24}$$

If we can solve the system above to get **Q**, then we can get $\tilde{\mathcal{U}}$ easily. Because the dimension of **Q** is much smaller than that of $\tilde{\mathcal{U}}$, we expect to get a reasonably fast algorithm for the two-phase Stokes equation if we can solve (3.24) efficiently.

In implementation, we use the GMRES [21] to solve (3.24). The GMRES method only requires the matrix vector multiplication. We explain below how to evaluate the right hand side $\bar{\mathbf{F}}$ of the Schur complement, and how to evaluate the matrix vector multiplication needed by the GMRES iteration. We can see why we do not need to form the coefficient matrix $E - SA^{-1}B$ explicitly.

3.1.1 Evaluation of the right hand side of the Schur complement

First we set $\mathbf{Q} = \mathbf{0}$ and solve the de-coupled system (3.17)-(3.19), or (3.21) in the discrete form, to get $\tilde{\mathcal{U}}(\mathbf{0})$ which is $A^{-1}\mathbf{F}_1$ from (3.21). From the interpolation scheme (3.22), we

also have

$$\left[\mathbf{U}(\mathbf{0})\right]\Big|_{\Gamma} = S\tilde{\mathcal{U}}\left(\mathbf{0}\right) + E\,\mathbf{0} - \mathbf{F}_2 = S\tilde{\mathcal{U}}\left(\mathbf{0}\right) - \mathbf{F}_2. \tag{3.25}$$

Note that the residual of the Schur complement for $\mathbf{Q} = \mathbf{0}$ is

$$\begin{aligned}
R(\mathbf{0}) &= (E - SA^{-1}B)\,\mathbf{0} - \bar{\mathbf{F}} = -\bar{\mathbf{F}} \\
&= -\left(\mathbf{F}_2 - SA^{-1}\mathbf{F}_1\right) = -\mathbf{F}_2 + S\tilde{\mathcal{U}}\left(\mathbf{0}\right) \\
&= \left[\mathbf{U}(\mathbf{0})\right]\big|_{\Gamma}
\end{aligned} \tag{3.26}$$

which gives the right hand side of the Schur complement system with an opposite sign.

3.1.2 Evaluation of the matrix-vector multiplication

The matrix-vector multiplication of the Schur complement system given $\mathbf{Q}$ is obtained from the following two steps:

Step 1: Solve the coupled system (3.17)-(3.19), or (3.21) in the discrete form, to get $\tilde{\mathcal{U}}\left(\mathbf{Q}\right)$.

Step 2: Interpolate $\tilde{\mathcal{U}}\left(\mathbf{Q}\right)$ using (3.22) to get $\left[\mathbf{U}(\mathbf{Q})\right]\big|_{\Gamma}$. Then the matrix vector multiplication is

$$\left(E - SA^{-1}B\right)\mathbf{Q} = \left[\mathbf{U}(\mathbf{Q})\right]\Big|_{\Gamma} - \left[\mathbf{U}(\mathbf{0})\right]\Big|_{\Gamma}. \tag{3.27}$$

This is because

$$\begin{aligned}
\left(E - SA^{-1}B\right)\mathbf{Q} &= E\mathbf{Q} - SA^{-1}B\mathbf{Q} \\
&= E\mathbf{Q} - S\left(A^{-1}\mathbf{F}_1 - \tilde{\mathcal{U}}\left(\mathbf{Q}\right)\right) && \left(\text{from (3.21)}\right), \\
&= E\mathbf{Q} + S\tilde{\mathcal{U}}\left(\mathbf{Q}\right) - \mathbf{F}_2 + \mathbf{F}_2 - SA^{-1}\mathbf{F}_1 \\
&= \left[\mathbf{U}(\mathbf{Q})\right]\big|_{\Gamma} + \bar{\mathbf{F}} && \left(\text{from (3.22)}\right), \\
&= \left[\mathbf{U}(\mathbf{Q})\right]\big|_{\Gamma} - \left[\mathbf{U}(\mathbf{0})\right]\big|_{\Gamma}, && \left(\text{from (3.26)}\right).
\end{aligned}$$

Now we can see that a matrix vector multiplication is equivalent to solving the coupled system (3.17)-(3.19), or (3.21) in the discrete form, to get $\tilde{\mathcal{U}}$, and using an interpolation scheme (3.22) to get $\left[\mathbf{U}(\mathbf{Q})\right]\big|_{\Gamma}$ at the control points.

Since we know the right hand side of the linear system of equation and the matrix-vector multiplication of the coefficient matrix, it is straightforward to use the GMRES or other iterative methods.

3.2 The least squares interpolation scheme to compute the residual.

The interpolation scheme (3.22) to evaluate $[\tilde{\mathbf{U}}/\mu]$ and $[\tilde{\mathbf{V}}/\mu]$ is crucial to the efficiency (accuracy and the number of iterations of the GMRES iteration) of the method. To reduce the number of iterations, it is important to couple the solutions on both sides of the interface using the jump conditions. Although the least squares interpolation scheme is not new idea anymore, the details vary with problems. We explain the least squares interpolation scheme for computing (3.22) to see why we have the second matrix-vector equation in expression (3.23).

Given an approximation to the augmented variable $\mathbf{Q}$, we can solve for the pressure, and then the velocity $(\mathbf{U}, \mathbf{V}) = (\tilde{\mathbf{U}}/\mu, \tilde{\mathbf{V}}/\mu)$ from (3.17)-(3.19). Since

$$\left[\frac{\tilde{\mathbf{U}}}{\mu}\right] = \frac{\tilde{\mathbf{U}}^+}{\mu^+} - \frac{\tilde{\mathbf{U}}^-}{\mu^-},$$

we need to evaluate $\{\tilde{U}^+\}$ and $\{\tilde{U}^-\}$ at all control points to get the vector $[\tilde{\mathbf{U}}/\mu]$. To explain the idea, however, we just need to explain the interpolation scheme for $\tilde{U}^-(\mathbf{X})$ at a point $\mathbf{X}$ on the interface. The interpolation scheme can be written as

$$\tilde{U}^-(\mathbf{X}) = \sum_{k=0}^{k_s-1} \gamma_k \tilde{U}_{i^*+k,\, j^*+k} + C, \tag{3.28}$$

where k_s is the number of grid points involved in the interpolation scheme, (x_{i^*}, y_{j^*}) is the closest grid point to $\mathbf{X}$, and C is a correction term. We should point it out that a one-sided interpolation scheme works poorly in the sense that the convergence speed is slow for the GMRES iteration. Below we discuss how to determine the coefficients γ_k and the correction term C using the information from both sides of the interface. Note that γ_k and C depend on $\mathbf{X}$. But for simplicity of the notation, we have omitted the dependency.

We use an un-determined coefficients method to determine the coefficients γ_k by minimizing the truncation error of (3.28) when $\tilde{U}_{i^*+k,\, j^*+k}$ is substituted by the exact solution $\tilde{u}(x_{i^*+k}, y_{j^*+k})$. Using the local coordinates system centered at the point $\mathbf{X}$, see (2.9), and denoting the local coordinates of (x_{i^*+k}, y_{j^*+k}) as (ξ_k, η_k), we have the following from the Taylor expansion:

$$\begin{aligned}
\tilde{u}(x_{i^*+k}, y_{j^*+k}) &= \tilde{u}(\xi_k, \eta_k) = \tilde{u}^\pm + \xi_k \tilde{u}_\xi^\pm + \eta_k \tilde{u}_\eta^\pm \\
&\quad + \frac{1}{2}\xi_k^2 \tilde{u}_{\xi\xi}^\pm + \xi_k \eta_k \tilde{u}_{\xi\eta}^\pm + \frac{1}{2}\eta_k^2 \tilde{u}_{\eta\eta}^\pm + O(h^3),
\end{aligned} \tag{3.29}$$

where the $+$ or $-$ sign is chosen depending on whether (ξ_k, η_k) lies on the $+$ or $-$ side of Γ, $\tilde{u}^\pm$, $\tilde{u}_\xi^\pm$, $\cdots$, $\tilde{u}_{\eta\eta}^\pm$ are evaluated at the local coordinates $(0,0)$, or $\mathbf{X} = (X, Y)$ in the original coordinates system. Note that we should have used something like $\tilde{u}(X, Y) = \bar{\tilde{u}}(0,0)$ to distinguish the two coordinate systems. However, we omit the bars and use the same notation $\tilde{u}(X, Y) = \tilde{u}(0,0)$ for simplicity.

We carry out this expansion at all the grid points used in the interpolation scheme and plug (3.29) into (3.28). After collecting and arranging terms, we can write

$$
\begin{aligned}
\tilde{u}^-(\mathbf{X}) \quad \approx \quad & a_1\, \tilde{u}^- + a_2\, \tilde{u}^+ + a_3\, \tilde{u}_\xi^- + a_4\, \tilde{u}_\xi^+ + a_5\, \tilde{u}_\eta^- + a_6 \tilde{u}_\eta^+ \\
& + a_7\, \tilde{u}_{\xi\,\xi}^- + a_8\, \tilde{u}_{\xi\,\xi}^+ + a_9\, \tilde{u}_{\eta\eta}^- + a_{10}\, \tilde{u}_{\eta\eta}^+ + a_{11}\, \tilde{u}_{\xi\eta}^- + a_{12}\, \tilde{u}_{\xi\eta}^- C,
\end{aligned}
\tag{3.30}
$$

where the a_i's are given by

$$
\begin{aligned}
& a_1 = \sum_{k \in K^-} \gamma_k && a_2 = \sum_{k \in K^+} \gamma_k \\
& a_3 = \sum_{k \in K^-} \xi_k \gamma_k && a_4 = \sum_{k \in K^+} \xi_k \gamma_k \\
& a_5 = \sum_{k \in K^-} \eta_k \gamma_k && a_6 = \sum_{k \in K^+} \eta_k \gamma_k \\
& a_7 = \frac{1}{2} \sum_{k \in K^-} \xi_k^2 \gamma_k && a_8 = \frac{1}{2} \sum_{k \in K^+} \xi_k^2 \gamma_k && \text{(3.31)} \\
& a_9 = \frac{1}{2} \sum_{k \in K^-} \eta_k^2 \gamma_k && a_{10} = \frac{1}{2} \sum_{k \in K^+} \eta_k^2 \gamma_k \\
& a_{11} = \sum_{k \in K^-} \xi_k \eta_k \gamma_k && a_{12} = \sum_{k \in K^+} \xi_k \eta_k \gamma_k.
\end{aligned}
$$

Note that $\tilde{u}^+ = \tilde{u}^- + q_1$ and $\left[\frac{\partial \tilde{u}}{\partial n}\right]$ is known from (2.11). From [12, 13], we also have the following interface relations[3]:

$$
\begin{aligned}
\tilde{u}_\eta^+ &= \tilde{u}_\eta^- + \frac{dq_1}{d\eta}, \\
\tilde{u}_{\xi\eta}^+ &= \tilde{u}_{\xi\eta}^- + \kappa \left[\frac{\partial \tilde{u}}{\partial n}\right] - \frac{d^2 q_1}{d\eta^2} + [\,p_x - g_1\,] \\
\tilde{u}_{\eta\eta}^+ &= \tilde{u}_{\eta\eta}^- + \kappa \left[\frac{\partial \tilde{u}}{\partial n}\right] + \frac{d^2 q_1}{d\eta^2}, \\
\tilde{u}_{\xi\eta}^+ &= \tilde{u}_{\xi\eta}^- + \kappa \frac{dq_1}{d\eta} + \frac{d}{d\eta}\left[\frac{\partial \tilde{u}}{\partial n}\right],
\end{aligned}
\tag{3.32}
$$

where κ is the curvature of the interface at $\mathbf{X}$. Therefore we can express all the quantities from $+$ side in (3.30) in terms of those from $-$ side and the known quantities. Thus (3.28),

---

[3]Note that q_1 now is a quantity defined only on the interface, and we can only take its derivatives along the interface which is called surface derivative in the literature.

when $\tilde{U}_{i^*+k,\,j^*+k}$ is substituted by the exact solution $\tilde{u}(x_{i^*+k},y_{j^*+k})$, can be written as

$$
\begin{aligned}
\tilde{u}^-(\mathbf{X}) &\approx \sum_k \gamma_k\,\tilde{u}(x_{i^*+k},y_{j^*+k}) + C\\
&= a_1\,\tilde{u}^- + a_2\,\tilde{u}^+ + a_3\,\tilde{u}_\xi^- + a_4\,\tilde{u}_\xi^+ + a_5\,\tilde{u}_\eta^- + a_6\tilde{u}_\eta^+ + a_7\,\tilde{u}_{\xi\,\xi}^- + a_8\,\tilde{u}_{\xi\,\xi}^+\\
&\quad + a_9\,\tilde{u}_{\eta\eta}^- + a_{10}\,\tilde{u}_{\eta\eta}^+ + a_{11}\,\tilde{u}_{\xi\eta}^- + a_{12}\,\tilde{u}_{\xi\eta}^+ + C\\
&= (a_1+a_2)\tilde{u}^- + (a_3+a_4)\tilde{u}_\xi^- + (a_5+a_6)\tilde{u}_\eta^- + (a_7+a_8)\tilde{u}_{\xi\xi}^-\\
&\quad + (a_9+a_{10})\tilde{u}_{\eta\eta}^- + (a_{11}+a_{12})\tilde{u}_{\xi\eta}^- + a_2[\tilde{u}] + a_4[\tilde{u}_\xi] + a_6[\tilde{u}_\eta]\\
&\quad + a_8[\tilde{u}_{\xi\xi}] + a_{10}[\tilde{u}_{\xi\eta}] + a_{12}[\tilde{u}_{\xi\,\eta}] + C.
\end{aligned}
$$

To minimize the local truncation error, we should set the following linear system of equations for the coefficients γ_k by matching the terms of $\tilde{u}^-,\,\tilde{u}_\xi^-,\,\cdots,\,\tilde{u}_{\xi\,\eta}^-$:

$$
\begin{aligned}
a_1 + a_2 &= 1,\\
a_3 + a_4 &= 0,\\
a_5 + a_6 &= 0,\\
a_7 + a_8 &= 0,\\
a_9 + a_{10} &= 0,\\
a_{11} + a_{12} &= 0.
\end{aligned}
\tag{3.33}
$$

The system of equations for $\{\gamma_k\}$ is independent of jumps which means we can calculate $\{\gamma_k\}$ outside of the GMRES iteration. Once we have the coefficients, the correction term is

$$
\begin{aligned}
C &= -\Big(a_2[\tilde{u}] + a_4[\tilde{u}_\xi] + a_6[\tilde{u}_\eta] + a_8[\tilde{u}_{\xi\xi}] + a_{10}[\tilde{u}_{\xi\eta}] + a_{12}[\tilde{u}_{\xi\,\eta}] \Big)\\
&= -a_2\,q_1 - a_4\left[\frac{\partial\tilde{u}}{\partial n}\right] - a_6\frac{dq_1}{d\eta} - a_8\left(\left[\frac{\partial\tilde{u}}{\partial n}\right]\kappa - \frac{d^2[\tilde{u}]}{d\eta^2} + [\,p_x - g_1\,]\right)\\
&\quad - a_{10}\left(\frac{d^2 q_1}{d\eta^2} - \left[\frac{\partial\tilde{u}}{\partial n}\right]\kappa\right) - a_{12}\left(\frac{d[\tilde{u}]}{d\eta}\kappa + \frac{d}{d\eta}\left[\frac{\partial\tilde{u}}{\partial n}\right]\right).
\end{aligned}
\tag{3.34}
$$

We choose a neighborhood of $\{\mathbf{X}\}$ that contains more than six different grid points so that we have an under-determined system. In our numerical tests, we choose $k_s = 12$, that is, we selected 12 closest grid points to $\mathbf{X} = (X,Y)$ as the interpolation stencil. We use the singular value decomposition (SVD) to find the least squares solution which also has the least l-2 norm among all the solutions. In this way, the magnitude of the coefficients γ_k is controlled and balanced. The least squares interpolation plays an important role in the stability of the algorithm.

The only trade-off of the least squares interpolation is that we have to solve an under-determined system of equations. However, the size of the linear system is small and the coefficients can be pre-determined before the GMRES iteration. The extra time needed in

dealing with the interface is usually less that 5% of the total CPU time and the percentage decreases as the mesh size (M and N) increases.

Remarks: By setting $a_1 + a_2 = 0$ and $a_3 + a_4 = 1$ while keeping other equations unchanged in (3.33), we can easily get the normal derivative of the solution u. This is the method that we used in Section 4 for the grid refinement analysis of the computed normal derivative of the velocity.

There are also a few issues about how to evaluate p_x and p_y at grid points near the interface. We also need to evaluate $[p_x]$ and $[p_y]$ at some points on the interface. We refer the reader to [12] for the detailed information. We just need to compute those quantities near or on the interface to first order accuracy since they appear at the right hand side of the Poisson equations.

4   Results and analysis of numerical experiments

We first present several grid refinement analysis for the examples in which we know the exact solutions to check the order of accuracy. Most of the computations are done on workstations or Laptop PC's within a few seconds to a few minutes for stationary Stokes problems. It is quite challenging to construct the exact solutions for incompressible Stokes flow with an interface. Throughout this section, the computational domain is $\Omega = [-2,\ 2] \times [-2,\ 2]$.

Example 4.1

We start with a simple example where the velocity is smooth and the pressure is discontinuous across the interface. The exact velocity and the pressure are given by

$$u = y\left(x^2 + y^2 - 1\right), \qquad (x,y) \in \Omega, \tag{4.35}$$

$$v = -x\left(x^2 + y^2 - 1\right), \qquad (x,y) \in \Omega, \tag{4.36}$$

$$p = \begin{cases} 1, & \text{if } x^2 + y^2 \le 1, \\ 0, & \text{if } x^2 + y^2 > 1. \end{cases} \tag{4.37}$$

The interface is the unit circle. The viscosity is

$$\mu = \begin{cases} 1, & \text{if } x^2 + y^2 \le 1, \\ \dfrac{1}{2}, & \text{if } x^2 + y^2 > 1. \end{cases} \tag{4.38}$$

The bounded external forcing term $\mathbf{g}$ is given by

$$g_1 = \begin{cases} -8y, & \text{if } x^2 + y^2 \le 1, \\ -4y, & \text{if } x^2 + y^2 > 1, \end{cases} \tag{4.39}$$

$$g_2 = \begin{cases} 8x, & \text{if } x^2 + y^2 \le 1, \\ 4x, & \text{if } x^2 + y^2 > 1, \end{cases} \tag{4.40}$$

which has a finite jump across the interface. The normal and tangential force density are

$$\hat{f}_1 = [p] - 2\left[\mu\frac{\partial \mathbf{u}}{\partial n}\cdot\mathbf{n}\right] = -1, \tag{4.41}$$

$$\hat{f}_2 = -\left[\mu\frac{\partial \mathbf{u}}{\partial n}\cdot\boldsymbol{\tau}\right] - \left[\mu\frac{\partial \mathbf{u}}{\partial \tau}\cdot\mathbf{n}\right] = -1, \tag{4.42}$$

calculated from (2.4) and (2.6) respectively.

In Table 1, we show the result of the grid refinement analysis. Since the exact solution is not periodic, we use the Dirichlet boundary condition when we solve the three Poisson equations (3.17)-(3.19) so that we can check the accuracy of the computed solution. This did not cause any inconsistence since we know that the solution exists. Similarly, the Dirichlet boundary condition is used for other examples with the exact solution that is not periodic in this section based on the same arguments. If we choose $\mathbf{0}$ as the initial guess, the GMRES iteration converges quickly since $\mathbf{0}$ is the exact solution of the augmented variables. Therefore we choose random numbers between 0 and 1 as the components of the initial guess to get a realistic evaluation of the algorithm. The tolerance for the GMRES iteration is taken as 10^{-6}. The errors in Table 1 are measured in the maximum norm at all grid points, for example

$$E_{\mathbf{u}}(N) = \frac{1}{2}\left(\max_{0\leq i,j\leq N}|U_{ij} - u(x_i, y_j)| + \max_{0\leq i,j\leq N}|V_{ij} - v(x_i, y_j)|\right), \tag{4.43}$$

where $u(x_i, y_j)$ is the exact solution at (x_i, y_j) while U_{ij} is the approximate solution and so on. In all the tables in this section, N is the number of grid lines in both x- and y- directions. The ratio

$$p\text{-order} = \frac{\log\left(E_p(N)/E_p(2N)\right)}{\log 2}, \tag{4.44}$$

for example, is an indication of the order of accuracy for the pressure. We can see roughly second order convergence for all the quantities. The last column is the number of iterations (No.) for the GMRES method. We can see that the number of iterations remains roughly the same as we double the grid lines in each direction.

Table 1: **Numerical results and convergence analysis for Example 4.1.**

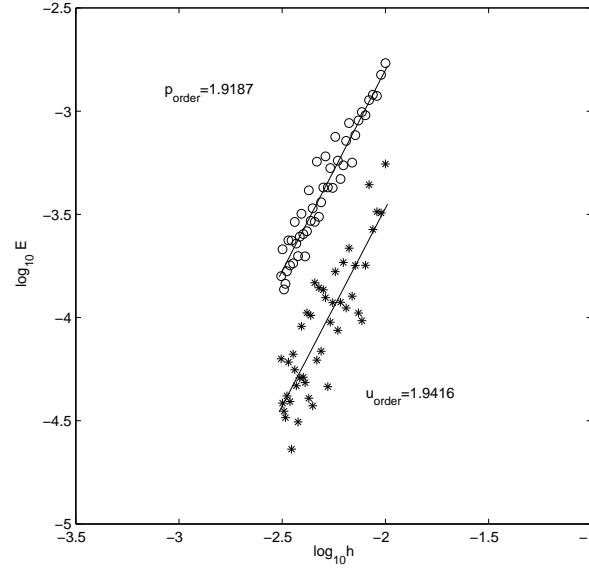| N	E_p	p-order	$E_{\mathbf{u}}$	$\mathbf{u}$-order	$E_{\partial\mathbf{u}/\partial n}$	$\frac{\partial\mathbf{u}}{\partial n}$-order	No.
32	8.2569×10^{-3}		6.5931×10^{-3}		2.4605×10^{-2}		10
64	3.0540×10^{-3}	1.4349	1.7372×10^{-3}	1.9242	6.0768×10^{-3}	2.0176	11
128	9.4745×10^{-4}	1.6886	3.9504×10^{-4}	2.1367	1.6713×10^{-3}	1.8623	12
256	2.6827×10^{-4}	1.8204	8.2274×10^{-5}	2.2635	4.1920×10^{-4}	1.9953	13
512	7.4298×10^{-5}	1.8523	2.5053×10^{-5}	1.7155	1.0445×10^{-4}	2.0048	14

Figure 2: Linear regression analysis for the pressure and the velocity for Example 4.1. The order of accuracy for the pressure and the velocity are 1.9187 and 1.9416 respectively.

Using fixed grids to solve an interface problem, the errors usually do not decrease monotonically as we refine the grid, see [14]. To be more precise, we use the linear regression analysis to find approximate order of accuracy. In Fig. 2, we show the error plot in log-log scale for the pressure and the velocity versus the grid spacing $h = h_x = h_y$, which shows that, from the slopes, the average convergence order of the pressure and the velocity are 1.9187 and 1.9416, respectively. The mesh size varies from $N = 100$ to $N = 320$ according to $N = 100 + 5k$, $k = 0, 1, \cdots, 44$. The order of accuracy for the normal velocity $\partial u / \partial n$ is 2.1928 from the linear regression analysis.

Example 4.2

In the first example, while the pressure is discontinuous, the velocity is smooth and vanishes at the interface. Therefore the exact solution to the augmented variable $\mathbf{q} = [\mu \mathbf{u}]$ is zero which may make the problem easier to compute. In this example, we keep all the solution inside the unit circle unchanged, but set both pressure and the velocity outside of the circle to be zero. Therefore, the periodic boundary conditions are satisfied. The velocity is non-smooth and the jump in the normal velocity is not a constant along the interface. Now the normal force density is still $\hat{f}_1 = -1$, but the tangential force density is

$$\hat{f}_2 = - \left[\mu \frac{\partial \mathbf{u}}{\partial n} \cdot \boldsymbol{\tau} \right] - \left[\mu \frac{\partial \mathbf{u}}{\partial \tau} \cdot \mathbf{n} \right] = -2. \tag{4.45}$$

We also have $\mathbf{g} = \mathbf{0}$ outside of the circle and there is a finite jump in $\mathbf{g}$ as well. In Table 2, we show the grid refinement analysis. We see again roughly average second order convergence

for all quantities. The number of iterations is small and does not change much. The results of the linear regression analysis for the pressure and the velocity are given in Fig. 3 which confirms average second order accuracy for both the pressure and the velocity.

Table 2: **Numerical results and convergence analysis for Example 4.2.**

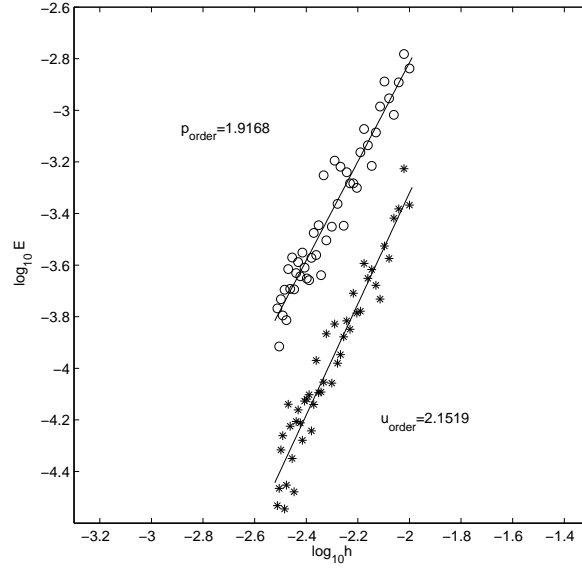| N	E_p	p-order	$E_{\mathbf{u}}$	$\mathbf{u}$-order	$E_{\partial\mathbf{u}/\partial n}$	$\frac{\partial\mathbf{u}}{\partial n}$-order	No.
32	8.4430×10^{-3}		3.4549×10^{-3}		2.8308×10^{-2}		9
64	2.8405×10^{-3}	1.5716	8.8800×10^{-4}	1.9600	6.1798×10^{-3}	2.1956	10
128	8.0952×10^{-4}	1.8110	2.2666×10^{-4}	1.9700	1.8260×10^{-3}	1.7589	11
256	2.5417×10^{-4}	1.6713	4.7693×10^{-5}	2.2487	5.3612×10^{-4}	1.7681	12
512	1.4086×10^{-5}	2.1296	1.4086×10^{-5}	1.7595	1.2538×10^{-4}	2.0962	13

Figure 3: Linear regression analysis for Example 4.2. The average convergence order for the pressure and the velocity are $p_{order} = 1.9168$, $u_{order} = 2.1519$.

Example 4.3

In previous examples, the force density are constants, and $[\mu\frac{\partial^2}{\partial\tau^2}(\mathbf{u}\cdot\mathbf{n})] = 0$. In this example, we construct the exact solutions in such a way that all the jumps and their derivatives along the interface are non-constant functions. The exact velocity and the pressure are given

by

$$
u = \begin{cases} \dfrac{y}{4}, & \text{if } x^2 + y^2 \le 1, \\[2mm] \dfrac{y}{4}\left(x^2 + y^2\right), & \text{if } x^2 + y^2 > 1, \end{cases} \tag{4.46}
$$

$$
v = \begin{cases} -\dfrac{x}{4}\left(1 - x^2\right), & \text{if } x^2 + y^2 \le 1, \\[2mm] -\dfrac{xy^2}{4}, & \text{if } x^2 + y^2 > 1, \end{cases} \tag{4.47}
$$

$$
p = \begin{cases} \left(-\dfrac{3}{4}x^3 + \dfrac{3}{8}x\right)y, & \text{if } x^2 + y^2 \le 1, \\[2mm] 0, & \text{if } x^2 + y^2 > 1. \end{cases} \tag{4.48}
$$

The bounded external forcing term **g** is

$$
g_1 = \begin{cases} \left(-\dfrac{9}{4}x^2 + \dfrac{3}{8}\right)y, & \text{if } x^2 + y^2 \le 1, \\[2mm] -2\mu^+ y & \text{if } x^2 + y^2 > 1, \end{cases} \tag{4.49}
$$

$$
g_2 = \begin{cases} -\dfrac{3}{4}x^3 + \dfrac{3}{8}x - \dfrac{3\mu^-}{2}x, & \text{if } x^2 + y^2 \le 1, \\[2mm] \dfrac{\mu^+}{2}x, & \text{if } x^2 + y^2 > 1, \end{cases} \tag{4.50}
$$

which is discontinuous across the interface. The force density corresponding to the singular Dirac delta function in the normal and tangential directions are

$$
\hat{f}_1 = \left(\dfrac{3}{4}\cos^3\theta - \dfrac{3}{8}\cos\theta\right)\sin\theta - \dfrac{3}{2}\left[\mu\right]\cos^3\theta\sin\theta, \tag{4.51}
$$

$$
\hat{f}_2 = \dfrac{1}{2}\mu^+ + \dfrac{3}{4}\left[\mu\right]\cos^2\theta\left(1 - 2\cos^2\theta\right), \tag{4.52}
$$

respectively. All the jump conditions (2.4)-(2.7) are satisfied except for (2.5). In our numerical test, we use a more general jump condition

$$
\left[\dfrac{\partial p}{\partial n}\right] = \left[\mathbf{g}\cdot\mathbf{n}\right] + \dfrac{\partial}{\partial\tau}\hat{f}_2 + 2\left[\mu\dfrac{\partial^2}{\partial\tau^2}(\mathbf{u}\cdot\mathbf{n})\right] + w. \tag{4.53}
$$

The added function w does not alter the difficulty and the nature of the problem and our algorithm but enables us to check the order of accuracy of the numerical method. In this example, w is given by

$$
w = 2\left[\mu\right]\cos^3\theta\sin\theta. \tag{4.54}
$$

We use the exact Dirichlet boundary condition for the pressure and the velocity.

In Table 3, we show the grid refinement analysis for different jump in μ. We scale the problem such that $\max\{\mu^-, \mu^+\} = 1$. We test our results for $\mu^-/\mu^+ = 10,\ 10^{-3}$, and 10^3. While the accuracy does depend on $[\mu]$, the average convergence rates are about the same (second order accurate). Note that, there are two very different scales for the problems in Table 3 (b) and Table 3 (c). The number of iterations seems to be dependent of the jump in μ but not the mesh size N.

Table 3: **Numerical results and convergence analysis for Example 4.3.**

(a) $\mu^- = 1$, $\mu^+ = 0.1$.

N	E_p	p-order	$E_{\mathbf{u}}$	$\mathbf{u}$-order	$E_{\partial \mathbf{u}/\partial n}$	$\frac{\partial \mathbf{u}}{\partial n}$-order	No.
32	1.1252×10^{-2}		1.0331×10^{-2}		1.2397×10^{-1}		10
64	3.1248×10^{-3}	1.8484	2.2832×10^{-3}	2.1779	2.5154×10^{-2}	2.3011	11
128	8.9338×10^{-4}	1.8064	5.5784×10^{-4}	2.0331	7.0234×10^{-3}	1.8405	10
256	2.4296×10^{-4}	1.8786	1.1291×10^{-4}	2.3047	1.4007×10^{-3}	2.3260	9
512	5.5515×10^{-5}	2.1297	2.8135×10^{-5}	2.0047	4.0508×10^{-4}	1.7899	8

(b) $\mu^- = 0.001$, $\mu^+ = 1$.

n	E_p	p-order	$E_{\mathbf{u}}$	$\mathbf{u}$-order	$E_{\partial \mathbf{u}/\partial n}$	$\frac{\partial \mathbf{u}}{\partial n}$-order	No.
32	1.4537×10^{-2}		9.9572×10^{-1}		1.3266		12
64	3.8694×10^{-3}	1.9095	2.2471×10^{-1}	2.1477	2.8381	2.2247	12
128	1.0974×10^{-3}	1.8180	5.7087×10^{-2}	1.9768	7.8753×10^{-1}	1.8495	11
256	3.4110×10^{-4}	1.6858	1.1255×10^{-2}	2.3426	1.8409×10^{-1}	2.0969	11
512	6.6231×10^{-5}	2.3646	2.8277×10^{-3}	1.9929	4.7873×10^{-2}	1.9431	9

(c) $\mu^- = 1$, $\mu^+ = 0.001$.

n	E_p	p-order	$E_{\mathbf{u}}$	$\mathbf{u}$-order	$E_{\partial \mathbf{u}/\partial n}$	$\frac{\partial \mathbf{u}}{\partial n}$-order	No.
32	2.8939×10^{-2}		2.2153		1.7133		27
64	5.4693×10^{-3}	2.4036	3.6865×10^{-1}	2.5872	2.5522×10^{-1}	2.7470	25
128	1.6101×10^{-3}	1.7642	9.1288×10^{-2}	2.0138	5.7709×10^{-2}	2.1449	25
256	4.1020×10^{-4}	1.9728	2.4121×10^{-2}	1.9201	1.4379×10^{-2}	2.0048	31
512	1.3470×10^{-4}	1.6066	6.7498×10^{-3}	1.8374	3.3450×10^{-3}	2.1039	30

Example 4.4

As a final test, we present an example of moving interface problem. The driven force is the surface tension, that is, the normal force is given by $\hat{f}_1 = \kappa$, where κ is the curvature of the interface Γ. The tangential force is zero. The interface Γ moves with the same velocity as the

fluid surrounding it, *i.e.*, $\frac{d\mathbf{X}}{dt} = \mathbf{u}$. The set-up is almost the same as the problem described in [12] except that the viscosity is now discontinuous. Since the emphasis of this paper is about the new method for stationary Stokes equations, we omit the details about the algorithm for the moving interface problems, but just show one simulation result. The initial interface in polar coordinates is set to be

$$\rho = 1 + 0.2\sin(5\alpha), \qquad 0 \le \alpha \le 2\pi. \tag{4.55}$$

The interface will move to its equilibrium, a circle. In Fig. 4, we show the initial interface and a computed approximation to the equilibrium. We tested three different cases with $(\mu^-, \mu^+) = (1, 0.01)$, $(1, 1)$, and $(0.01, 1)$. Fig. 5 shows the close-up plot of the computed interface at $t = 0.5$ for the three different cases. As it is expected that the interface moves faster if the viscosity is small. How the jump in the viscosity affects the motion of the interface is still under investigation.
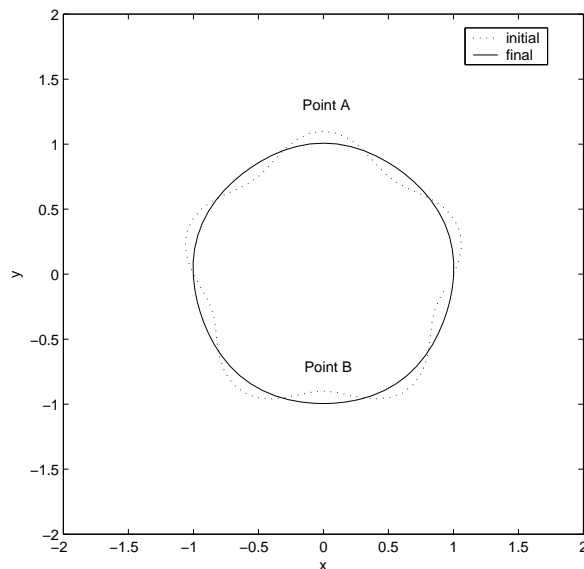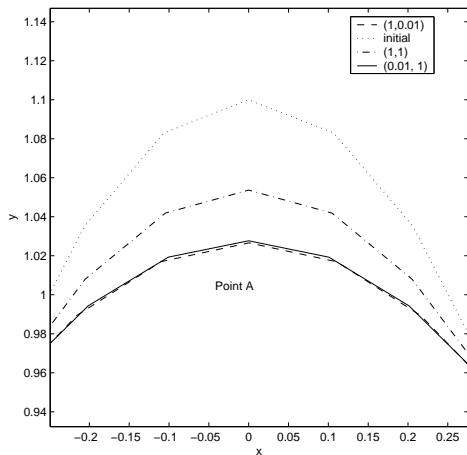

Figure 4: Plots of the initial interface (the dotted line) and the final equilibrium (the solid line) of Example 4.4.

Conclusion and acknowledgments

In this paper, a new second order accurate numerical method has been developed for incompressible stationary Stokes equations with discontinuous viscosity in which the jump conditions for the pressure and the velocity are coupled together. The idea is to introduce two augmented variables that are only defined along the interface so that the jump conditions can be decoupled. The GMRES iterative method then is used to solve the Schur complement
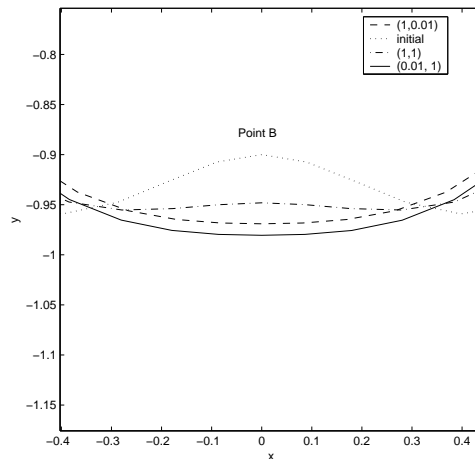
(a). (b).

Figure 5: Plots of the computed interface near the two points A and B with different viscosity ratio at $t = 0.2$. The dotted line is the initial interface. The dash-dotted line is the result with $(\mu^-, \mu^+) = (1, 1)$. The solid line is the result with $(\mu^-, \mu^+) = (0.01, 1)$. The dashed line is the result with $(\mu^-, \mu^+) = (1, 0.01)$.

system for the augmented variables. The main cost in one step GMRES iteration is about three calls to a fast Poisson solver. Numerical examples shown the efficiency of the method in accuracy and the speed. We believe that the new method may be the first second order sharp interface method for Stokes flow with discontinuous viscosity. The idea should be applicable to other interface problems with coupled jump conditions.

References

[1] G. Biros, L. Ying, and Z. Denis. An embedded boundary integral solver for the Stokes equations. TR2003-837, NYU, 2002.

[2] G. Chen, Z. Li, and P. Lin. A fast finite difference method for biharmonic equations on irregular domains. CRSC-TR04-09, North Carolina State University, 2004.

[3] R. Cortez. The method of regularized Stokeslets. *SIAM J. Sci. Comput.*, 23:1204–1225, 2001.

[4] A. L. Fogelson and C. S. Peskin. Numerical solution of the three dimensional Stokes equations in the presence of suspended particles. In *Proc. SIAM Conf. Multi-phase Flow*. SIAM, June 1986.

[5] L. Greengard, M. C. Kropinski, and A Mayo. Integral equation methods for Stokes flow and isotropic elasticity in the plane. *J. Comput. Phys.*, 125:403–414, 1996.

[6] T. Hou, Z. Li, S. Osher, and H. Zhao. A hybrid method for moving interface problems with application to the Hele-Shaw flow. *J. Comput. Phys.*, 134:236–252, 1997.

[7] J. Hunter, Z. Li, and H. Zhao. Autophobic spreading of drops,. *J. Comput. Phys.*, 183:335–366, 2002.

[8] K. Ito, R. LeVeque, and Z. Li. Interface relations for Stokes equations with discontinuous viscosity and singular sources. preprint, 2004.

[9] M. Kang, R. Fedkiw, and X. Liu. A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput*, 15:323–360, 2000.

[10] M-C. Lai and Z. Li. A remark on jump conditions for the three-dimensional Navier-Stokes equations involving an immersed moving membrane. *Applied Math. Letters*, 14:149–154, 2001.

[11] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31:1019–1044, 1994.

[12] R. J. LeVeque and Z. Li. Immersed interface method for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 18:709–735, 1997.

[13] Z. Li. *The Immersed Interface Method — A Numerical Approach for Partial Differential Equations with Interfaces*. PhD thesis, University of Washington, 1994.

[14] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35:230–254, 1998.

[15] Z. Li. An overview of the immersed interface method and its applications. *Taiwanese J. Mathematics*, 7:1–49, 2003.

[16] Z. Li and C. Wang. A fast finite difference method for solving Navier-Stokes equations on irregular domains. *J. of Commu. in Math. Sci.*, 1:180–196, 2003.

[17] Z. Li, H. Zhao, and H. Gao. A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid. *J. Comput. Phys.*, 152:281–304, 1999.

[18] A. Mayo and C. S. Peskin. An implicit numerical method for fluid dynamics problems with immersed elastic boundaries. *Contemp. Math.*, 141:261–277, 1991.

[19] C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25:220–252, 1977.

[20] C. S. Peskin. Lectures on mathematical aspects of physiology. *Lectures in Appl. Math.*, 19:69–107, 1981.

[21] Y. Saad. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[22] R. Teman. *Navier-Stokes Equations.* North-Holland, New York, 1977.

[23] C. Tu and C. S. Peskin. Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods. *SIAM J. Sci. Stat. Comput.*, 13:1361–1376, 1992.